

# Open Systems Engineering Effectiveness Measurement'

Presented at the Software Technology Conference  
by

Norman W. Kowalski, James T. Oblinger Jr., William J. Peresta

Naval Undersea Warfare Center Division Newport  
Code 2233, Newport, Rhode Island, 02841

Track Number: 9, Tuesday 21 April 1998

Keywords: Navy; effectiveness measurement; open systems engineering;

## Abstract:

This paper explores open systems engineering effectiveness measures and how they might be applied to programs undertaking an Open Systems approach. It will address ways in which programs, sponsoring and contracting agencies, and system integrators and developers may consider the effectiveness of their open systems engineering efforts. As with other engineering measurement activities, measuring the effectiveness of an open systems engineering effort, provides a means to objectively identify and manage risk, provides indications of potential problems, and provides a basis for informed decision making and communication. The paper specifically addresses:

- The underlying engineering and business premises that form the basis for an open systems engineering effort;
- The end goals and associated measurements of progress toward the end goals in a time phased perspective;
- Constraints on the potential success of an open systems engineering approach;
- Open systems measurement categories: management, process, product

## 1. Introduction

This paper explores open systems engineering effectiveness measures and how they might be applied to programs undertaking an Open Systems approach. This paper will address ways in which programs, sponsoring and contracting agencies, and system integrators and developers may consider the effectiveness of their open systems

---

<sup>1</sup> This work was supported in part by the Engineering of Complex System Program funded by the Office of Naval Research (ONR) and managed by the Naval Surface Weapons Center Dahlgren Division (NSWCDD)

## REPORT DOCUMENTATION PAGE

<b>1. REPORT DATE (DD-MM-YYYY)</b> 21-04-1998	<b>2. REPORT TYPE</b>	<b>3. DATES COVERED (FROM - TO)</b> xx-xx-1998 to xx-xx-1998
<b>4. TITLE AND SUBTITLE</b> Open Systems Engineering Effectiveness Measurement  Unclassified	<b>5a. CONTRACT NUMBER</b>	
	<b>5b. GRANT NUMBER</b>	
	<b>5c. PROGRAM ELEMENT NUMBER</b>	
<b>6. AUTHOR(S)</b> Kowalski, Norman W. ; Oblinger, James T. ; Peresta, William J. ;	<b>5d. PROJECT NUMBER</b>	
	<b>5e. TASK NUMBER</b>	
	<b>5f. WORK UNIT NUMBER</b>	
<b>7. PERFORMING ORGANIZATION NAME AND ADDRESS</b> Naval Undersea Warfare Center Division Newport Code 2233 Newport , RI 02841	<b>8. PERFORMING ORGANIZATION REPORT NUMBER</b>	
<b>9. SPONSORING/MONITORING AGENCY NAME AND ADDRESS</b>  ,	<b>10. SPONSOR/MONITOR'S ACRONYM(S)</b>	
	<b>11. SPONSOR/MONITOR'S REPORT NUMBER(S)</b>	
<b>12. DISTRIBUTION/AVAILABILITY STATEMENT</b> A PUBLIC RELEASE   ,		

**13. SUPPLEMENTARY NOTES****14. ABSTRACT**

This paper explores open systems engineering effectiveness measures and how they might be applied to programs undertaking an Open Systems approach. It will address ways in which programs, sponsoring and contracting agencies, and system integrators and developers may consider the effectiveness of their open systems engineering efforts. As with other engineering measurement activities, measuring the effectiveness of an open systems engineering effort, provides a means to objectively identify and manage risk, provides indications of potential problems, and provides a basis for informed decision making and communication. The paper specifically addresses: 1 The underlying engineering and business premises that form the basis for an open systems engineering effort; 1 The end goals and associated measurements of progress toward the end goals in a time phased perspective; 1 Constraints on the potential success of an open systems engineering approach; 1 Open systems measurement categories: management, process, product

**15. SUBJECT TERMS**

"IATAC COLLECTION"

<b>16. SECURITY CLASSIFICATION OF:</b>			<b>17. LIMITATION OF ABSTRACT</b> Public Release	<b>18. NUMBER OF PAGES</b> 29	<b>19a. NAME OF RESPONSIBLE PERSON</b> Fenster, Lynn lfenster@dtic.mil
<b>a. REPORT</b> Unclassified	<b>b. ABSTRACT</b> Unclassified	<b>c. THIS PAGE</b> Unclassified			<b>19b. TELEPHONE NUMBER</b> International Area Code  Area Code Telephone Number 703 767-9007 DSN 427-9007

<b>REPORT DOCUMENTATION PAGE</b>			Form Approved OMB No. 074-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing this collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.				
<b>1. AGENCY USE ONLY (Leave blank)</b>		<b>2. REPORT DATE</b> 4/22/98	<b>3. REPORT TYPE AND DATES COVERED</b> Report	
<b>4. TITLE AND SUBTITLE</b> Open Systems Engineering Effectiveness Measurement			<b>5. FUNDING NUMBERS</b>	
<b>6. AUTHOR(S)</b> Norman W. Kowalski, James T. Oblinger, Jr., William J. Peresta				
<b>7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)</b> IATAC Information Assurance Technology Analysis Center 3190 Fairview Park Drive Falls Church VA 22042			<b>8. PERFORMING ORGANIZATION REPORT NUMBER</b>	
<b>9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)</b> Defense Technical Information Center DTIC-IA 8725 John J. Kingman Rd, Suite 944 Ft. Belvoir, VA 22060			<b>10. SPONSORING / MONITORING AGENCY REPORT NUMBER</b>	
<b>11. SUPPLEMENTARY NOTES</b>				
<b>12a. DISTRIBUTION / AVAILABILITY STATEMENT</b>			<b>12b. DISTRIBUTION CODE</b>  A	
<b>13. ABSTRACT (Maximum 200 Words)</b>				
<b>14. SUBJECT TERMS</b> Metrics			<b>15. NUMBER OF PAGES</b>	
			<b>16. PRICE CODE</b>	
<b>17. SECURITY CLASSIFICATION OF REPORT</b> Unclassified	<b>18. SECURITY CLASSIFICATION OF THIS PAGE</b> UNCLASSIFIED	<b>19. SECURITY CLASSIFICATION OF ABSTRACT</b> UNCLASSIFIED	<b>20. LIMITATION OF ABSTRACT</b>  None	

engineering efforts. The focus of this open systems measurement paper is on measuring the effectiveness of an ongoing open systems engineering effort during the effort, in order to:

- Provide feedback on the progress of the effort;
- Provide a means to objectively identify and manage risk;
- Provide indications of potential problems, and;
- Provide a basis for informed decision making and communication.

This paper does not identify all of the associated progress measurements that are appropriate to consider for a comprehensive open systems engineering approach. Rather, concepts are explained, associated with underlying principles, that can be used to develop a full range of measures that are program specific. An open systems engineering approach is considered to be a supplement to programmatic and systems engineering efforts within a program. Such systems engineering efforts include requirements analysis, functional analysis and allocation, design synthesis and verification, and system analysis and control. [1]

The work explained within this paper is expected to be continued, and will be added to a currently available public home page entitled "Practical Open Systems Engineering" (POSE) currently maintained by the Naval Undersea Warfare Center Division Newport at URL: <http://arch6.npt.nuwc.navy.mil/pose/>. [2]

## **2. An Open Systems Engineering Perspective**

The Department of Defense faces significant challenges in implementing its policies [1,3,4] on use of commercial standards and products in military systems. DoD military systems' developers once controlled all aspects of product development, and now are faced with integrating commercial technologies and products into their systems. Commercial products and technologies have significantly different perspectives than those of the military developed products and technologies. They differ with respect to areas such as product life, the duration of product support, and the amount of training and documentation available for a product. This is because the commercial world faces challenges from competitors that preclude its being able to tailor its products to the long duration deployment preferences of the military world, and the specific configurations of product use on specific military platforms. While the commercial world would like to please each of its customers, it must focus on the meeting the needs of large customers while facing the challenges of a fierce competitive market.

One of the reasons for moving toward use of commercial technology is to take advantage of the explosive progress being made in areas such as information technology. While the controlled military development activities of the past provided significant stability, in areas such as product life and duration of product support, it also produced a relatively stagnant environment in which change could also only be introduced slowly and with great effort. Successfully merging the stability (predictability in performance,

reliability, maintainability, ...) of deployed systems within the military with the technology innovation of the commercial industry is the challenge of this paradigm.

An open systems approach, a mandated policy of the DoD [1] has the potential to contribute to the successful merger of the military and commercial worlds, but is a relatively new engineering entity. There are a number of perspectives [5,6,7,8] on what an open systems engineering approach is, and how it should be undertaken.

An open systems engineering effort is an architecture effort undertaken to provide a resilient infrastructure. An open systems approach has the potential to significantly reduce the risks associated with the use of commercial products in mission critical military systems both technically and economically. The resiliency (durability) provides the means to adapt to a changing environment. As the commercial world changes, the standards based approach provides a stable baseline into which evolving technologies can be integrated. Facilitating integration is the means by which economic benefits are achieved.

Such benefits accrue by being able to rapidly and efficiently integrate upgrades and system component changes; being able to port or move applications in software and hardware to different vendors' platforms; and, being able to interoperate with other systems through standard interface protocols. Adopting commercial technology avoids component development costs. Adopting an open systems approach allows the mitigation of costs associated with commercial marketplace volatility because of the resiliency of the commercial interface standards based architecture.

There are many good characterizations of an open system, which reflect economic goals. Terms such as portability, interoperability, scalability, vendor independence, and supportability are achieved through engineering to meet various economic rather than performance related characteristics of a system development effort. Portability, for example, addresses being able to reduce cost and schedule efforts associated with moving some functionality (usually software) to another platform. Without adherence to standards, the ability to port software to platforms for which it was not originally intended is a very costly effort. The act of developing software that targets standards based platforms provides a greater number of platforms on which an application is likely to operate. Interoperability addresses the ability of two dissimilar systems to exchange and use information that has been exchanged. The network paradigm is a good example of interoperability. If all systems used unique interfaces, each system attempting to communicate to other systems would have to develop a unique interface for the system it wanted to inter-operate with [9].

There are a number of very good reference sources for formal definitions of and information on open systems engineering and associated concepts. The DoD sponsored Open Systems Joint Task Force [S] maintains a home page with a number of substantive definitions and resources identified.

For the purposes of understanding this paper a particular set of definitions is used. The definition of an open system that will be used, is that endorsed by the DoD [S] and adapted from the IEEE [10]. This definition is as follows:

“An Open System is a system that implement sufficient open specifications for interfaces, services and supporting formats to enable properly engineered components to be utilized across a wide range of components with minimal changes, to inter-operate with other components on local and remote systems, and to interact with users in a style that facilitates user portability.”

Variations of this definition exist, which are probably equally valid. A key element of this definition is the phrase properly engineered components. This phrase correlates well with the engineering process activities necessary to successfully achieve an open system.

A key open systems engineering concept is that of an open standard interface profile. Since open commercial standards contain required (mandatory) features, optional features and implementation configurable features, a profile is used to describe those features designated for use within a system. In an ideal world, any product claiming conformance to an interface standard, must meet the mandatory requirements of the interface standard. An interface profile is used to describe which options and implementation configurable features of a standard are desired (requirements specification) or implemented (implementation specification). [11].

A system profile selects base standard interface requirements, options, and other implementation configurable parameters to be applied to a system. A systems profile is normally a group of interface profiles that are appropriate for a system. The goal of defining a system profile is to provide a complete and coherent subset of an open system environment, which supports the open system goals for a system's constraints and performance requirements. From an open system perspective, a profile does not restrict components to be the same except at the interface level. One could consider a profile to be a very specific subset of standards to be applied within a given environment. [8]

Conformance is yet another key open systems engineering term. Conformance refers to the development of components that meet the interface specifications designated in an open standards interface standard or in a system profile.

Implementation conformance refers to vendor offered products (Commercial-Off-The-Shelf (COTS) products) that conform to a specific profile of standard features. Vendors often provide non-conforming features within their implementations as one way of differentiating their products from the products of other vendors.

Application conformance refers to user-developers of the system ensuring that only profile conformant interface “calls” are used to access the implementation, and not

using the vendor provided additional features. Standards bodies provide very strict and detailed definitions of conformance that need to be addressed when developing open systems [12]. A profile is normally a smaller subset of the standards interface features [8]. From the standard conformant features, a specific set (profile) is designated for use with applications, which, if adhered to, results in a conforming application.

## **2.1 Open System Characteristics**

Two open systems may be very different in function, interfaces used, standards, yet both may yield open system benefits. Two different open systems may use the same interface standards differently, but both yield open system benefits appropriate to their system economic goals. Determining which issues, open system goals and economic benefits are to be addressed is an important part of tailoring an open systems approach to achieve them. If, for example, a subsystem needs only to exchange limited data with other subsystems, emphasis on network interfaces is an appropriate approach. If portability of software is the goal, a different engineering emphasis is important.

At a theoretical level the perspectives of all open system efforts are fundamentally the same. The similarities derive from open commercial standard interfaces. A simplified description of the importance of interfaces to an open systems approach is that use of standard interfaces provides a stable engineering framework into which many different products may be used. The interface standards that apply to light bulb sockets and electrical outlets are examples of this. Competitive products from different vendors “plug and play” into these electrical standards. Figure 1 illustrates open commercial technology standards as having similar longevity and stability, able to support products from different vendors as well as different generations of products. The stability of the interfaces is one key to addressing relative turbulence of the commercial product world as it is used in **military** systems.



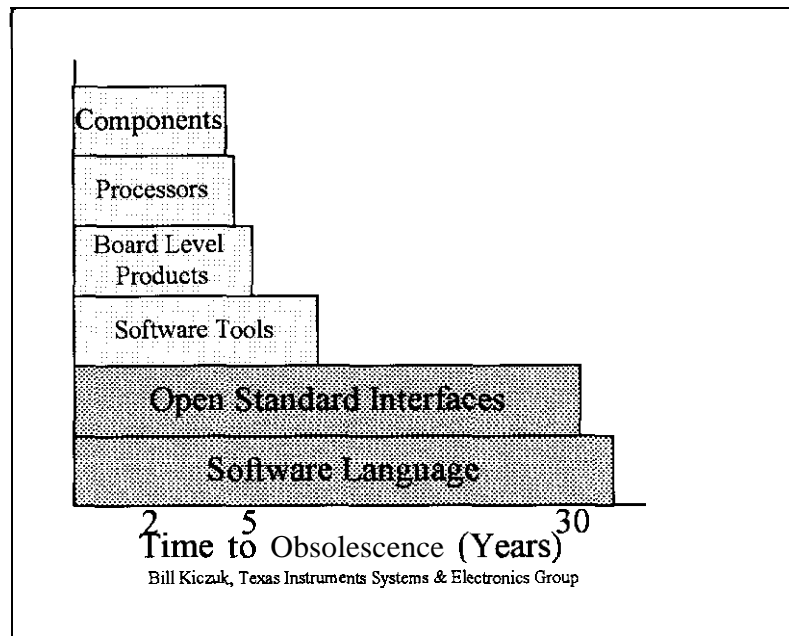


Figure 1 - Open Standard Interfaces - “a stabilizing factor”

## 2.2 Open Systems Engineering Measurement Dimensions

While it remains important to choose products which conform to open interface standards, information technology interface standards are more complex and relatively immature in comparison to these simple electrical examples. Plug and play is not generally available in information technology areas with the same level of assurance as for electrical outlets. If the premise that: *plug and play* is not automatic, is accepted, then some form of engineering must be necessary. That is, understanding product constraints, specifying how a product is to be used in order to not compromise the standard interface integrity, while not changing either the product or interface are simple examples of the engineering needed to effectively “plug and play” an information technology product.

Implicit in these engineering activities are management and engineering process based activities. One of the fundamental premises of this paper is that in order to achieve an open system, there is need for the successful application of:

- Management activities to facilitate through budgeting, contracting, and other actions;
- Engineering processes which provide consistent, well understood, repeatable procedures and activities in developing systems; and,
- Open commercial standard conformant products meeting designated requirements and system specific standard’s profiles.

In measuring the progress of an open systems engineering approach, it will be shown within this paper that the contributions of each of these must be measured. Figure 2 is meant to imply that the effectiveness of an open systems engineering approach is

multi-dimensional, represented by the volume of the cube and is not a simple, single dimensional measure.

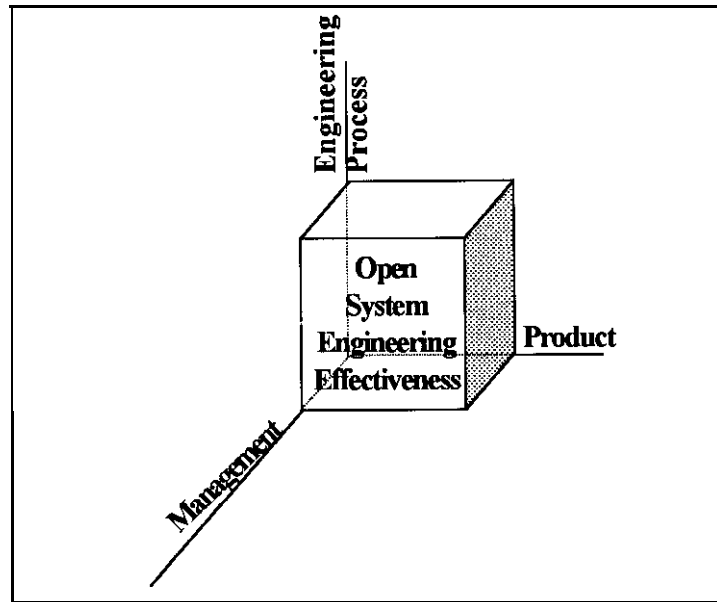


Figure 2 - Open Systems Engineering Effectiveness

It will be shown that this multi-dimensional aspect of open systems precludes the validity of claiming openness simply because one is using “open” products in the development of a system. If the “open” products are used inappropriately, the overall effectiveness of the entire effort may be significantly diminished. The emphasis of this paper is on measuring the progress of an open systems engineering approach within a program on all three dimensions; engineering process, product and management. The progress assessment approach provides insight into risk and problem areas and provides a basis for informed decision making and communication during system development. This approach for open systems engineering measurement follows the philosophy of the software measurement initiatives of the Practical Software Measurement program [13].

Another aspect of open systems engineering which must be considered is that success in each of the areas of management, engineering process and product, is highly dependent on the phase of the project. That is, if a project/system is considered to have three phases: design, development and deployment/sustainment, successful management activities in the design phase do not automatically imply successful management activities for all phases of the project. For this paper, these three phases are used to illustrate the phased dependency of open systems engineering measurement. Planning and budgeting activities are assigned to the design phase, for the purposes of this paper. Different measurement criteria apply at different phases of a project. Hence progress may be assessed as being different, dependent upon the phase of the project. For example, a project that has successfully specified the use of open standards profiles in a procurement, during the design phase, but does not follow through and complete the profiles, and then

use the profiles in guiding system development, is not likely to succeed in achieving open system engineering goals. The measurement activity would indicate the differentiation between the levels of success achieved in the different phases. Different measurement criteria are applied to the different phases of a project so that insight into progress appropriate to the particular phase can be understood. Another consideration is that the ability to succeed in a subsequent phase may be dependent upon the level of success achieved in a prior phase. For example, if profile activity is poorly budgeted for and the consequent profile is poorly developed, the ability to achieve a profile application will be severely impaired.

Since the measurement activity can be different during each system phase, it is reasonable to view each phase with its own system measurement cube. Figure 3a implies a differently shaped cube for each of the Design, Development and Sustainment phases. On each axis is a mark that denotes an optimal measure that may be obtainable for that category during that system phase. Management is depicted as being more important during the system design phase, where policies and direction may be most critical, than the development phase. Management rises again in the Sustainment phase where managing technical refresh and upgrade become more important. This figure currently shows only possible relative qualitative measures and not quantitative measures and is only intended as a notional aid

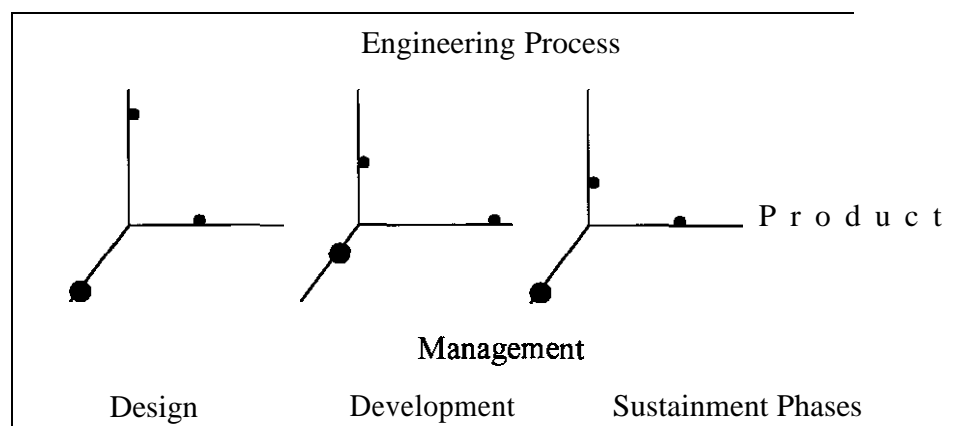


Figure 3a - Open Systems Engineering Effectiveness and Life Cycle Phase

Figure 3b shows the same data as Figure 3a but emphasizes the variation in the importance of the three categories over the system life cycle. What does not show explicitly in these figures but is important to remember is that there is linkage between the phases. If you do a poor management job during the Design phase, you will be unable to successfully achieve OSA success during the Development phase.

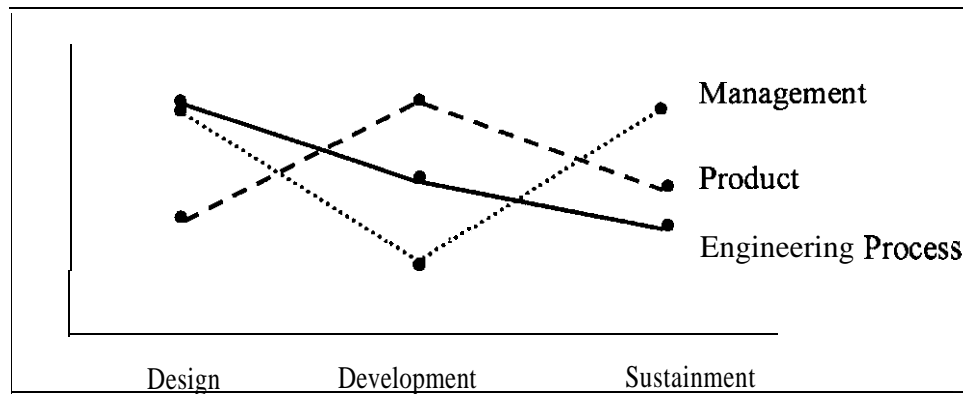


Figure 3b - Open Systems Engineering Effectiveness and Life Cycle Phase

### 3.0 Issues

There are a number of significant issues facing the DoD community in building, operating and maintaining mission critical systems, composed of commercial components and using open standards. Some of the issues are common to all programs, while others reflect the particular requirements of specific programs. Since open systems engineering is a nascent discipline, there is sometime confusion on why certain open systems engineering activities are undertaken and what contributions they make to the overall program. This section provides a simple business reference model and identities engineering issues that address different aspects of the business model.

#### 3.1 Business Model

When a project is viewed from a purely business perspective, there are just three basic high level objectives that are to be meet. These three objectives are: 1) Can the system be completed within schedule, 2) within budget, and 3) are the risks associated with the system development understood and manageable.

These three objectives, and more accurately they could be called stresses, act upon the developing activity, and determine many of the business and engineering decisions that are made. A new and unforeseen risk that appears may threaten the project schedule and budget. Likewise one project activity overspending its schedule or dollar budget may threaten the budgets of other project activities.

All these considerations are well known. What is important to this paper is to understand that there is a correlation between the business model and the engineering model. Engineering issues are derived from Business model objectives and interactions. Likewise business issues must be sensitive to Engineering issues.

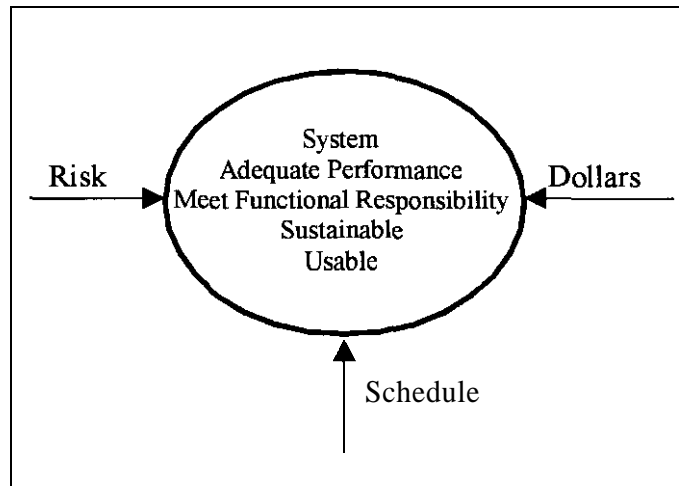


Figure 4: Business Objectives and Stresses

### 3.2 Engineering Issues

The goals of an open systems engineering approach identified in Section 4 of this paper were derived from the issues associated with developing a system under the conditions of today's DoD environment. Some of the issues are discussed below.

#### **Mission critical weapons systems need system stability but are composed of volatile commercial products.**

A major issue for the military is to merge the stability (predictability in performance, reliability, maintainability, . . .) of military systems and the technology innovation of the commercial industry. When military systems were developed completely under the control of a program, using strict military standards, and building products tailored to the specific requirements of the program, the strict controls provided the means to achieve the degree of stability appropriate to the system being developed. Commercial products are built to serve a wide range of users, and must compete with similar products from other vendors. Commercial product vendors cannot generally tailor their products to meet the needs of each individual program, and can only be responsive to large users. An open systems engineering approach, specifically through the use of standards provides a means to exert a level of control over the interfaces used in system development, and the associated processes. Goals to address this issue are included within this paper.

## **Commercial component compatibility and interoperability is difficult to attain and retain under market-driven conditions.**

A related issue is that of commercial component compatibility and interoperability. As subsystems are developed independently of one another, there is a strong likelihood that different component vendor's parts will be chosen for different subsystems. Eventually subsystems will be required to integrate to one another. As vendor products change throughout the life cycle of their deployment within a system, how can a deployed system be assured that replacement parts will work properly with the configurations of equipment deploying precedent versions of the part? Where staying with the same vendor provides some stability and assurance of compatibility, this is not always possible. One goal of an open systems engineering approach is to provide the means to minimize the effort associated with parts replacement, system integration and systems' interactions.

## **Becoming dependent on a particular commercial product can cause a loss of price-performance choices, which remains a major premise of using commercial standard products.**

One of the drawbacks of undisciplined commercial product approach is the single vendor dependence. Innovation, additional performance, and different capabilities being offered by other vendors could not easily be transitioned to. The DoD mandate to use open commercial standard based products alleviates the issue, because different vendor products meeting the same standard can be used based on the price-performance requirements of a program. However, again, "plug and play" is not yet a reality and there is engineering effort required in order to take advantage of alternate vendors products. The open systems engineering approach addresses this issue.

In choosing a commercial product for a system, one of the traps to avoid is getting "locked into" that vendor for life, producing a similar situation to that of a militarized approach. That particular product line may not suffice over time. Open systems engineering provides the means to address this issue by invoking engineering processes that limit vendor dependence and provide alternative engineering opportunity.

## **Transition to improved technology**

One of the issues associated with the rapid pace of commercial product change is to be able to insert a newer or updated technology at a reasonable cost. The rates of innovation, product change, and availability affect the supportability of "older" products. As newer products arrive, performance boosts for systems can also be important. It is not practical from a program perspective to transition from one technology to another, unless a means exists to do so in an affordable manner. The vendors that are providing the

products recognize the need to provide product that can be transitioned to also. If they meet standards, then it is easier for other users to transition to them. An open standards engineered system can accommodate technology change through the use of stable interfaces being used by changing technologies.

### **The integration of complex systems is difficult to manage (schedule, budget)**

The effort associated with integrating subsystems and components with uniquely designed interfaces is extremely **difficult** to justify today for many technical areas. The use of standards based interfaces, reduces the interdependency of subsystem or component development. Consequently, a more predictable integration schedule can be achieved, a shorter integration time and consequently a more timely introduction of a system can be achieved.

### **Cost Minimization over the life-cycle**

The DoD cannot afford to build and maintain military-unique systems, and has adopted the use of commercial technology as a cost-minimizing alternative. While use of commercial products does minimize development costs, maintenance costs where rapidly changing commercial products can cause havoc with respect to managing a configuration. The use of open systems engineering contributes to managing the interfaces by providing a stable control mechanism into which products integrate.

## **4.0 Goal-Question-Metric Format**

The measurement activities of this paper are presented to reader in Goal-Question-Metric format (Figure 5) suggested by Sage [14]. This method states a goal, reposes the goal as a number of questions or statements of requirements associated with the goal, and then applies suitable measures to gauge the extent to which the question has been answered. An issue-driven approach was used to derive the goals. The goals are based on issues that are faced in developing mission critical systems using commercial products.

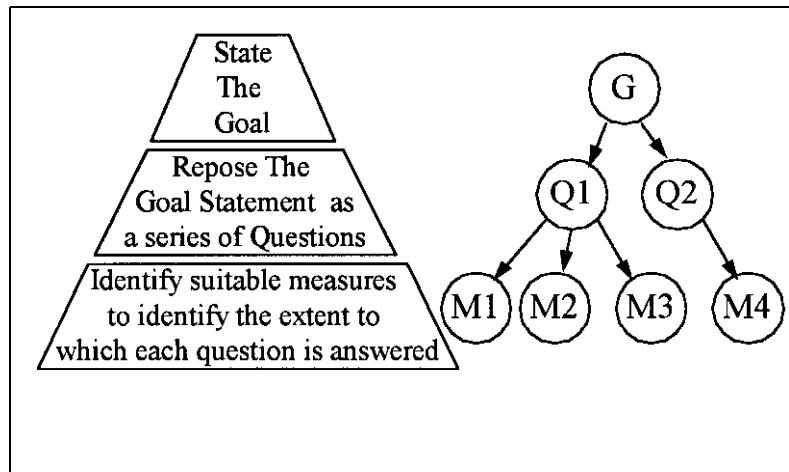


Figure 5 - Coal-Question-Measure format

Figure 6 illustrates the phased nature of an open systems engineering approach with respect to measurement. In this figure, a general business goal (Cost Management over the life cycle) is related to an open system engineering activity goal (Application Portability). This relationship is explained in detail in later parts of the paper, but for the purpose of explaining Figure 4, it should be understood that there is a direct relationship between cost management and building portable applications. Given these goals, a series of engineering questions (Q1, Q2, Q3,) are asked with respect to building portable applications. These questions lead to measurement activities that must take place during different phases of building the system. Figure 4 shows that measures M1 and M2, apply to question Q1 and are appropriate to address in the Design Phase. M3 applies to Q2. M4 applies to Q3. M3 and M4 are also important to address in the design phase. M4 is applied in all of the phases as illustrated.

Given the lack of engineering and management experience in developing open systems, one of the more difficult areas to address is the relationship of business issues and open systems engineering goals. The simple illustration of Figure 4 shows application portability relating to cost management. During the course of this paper, a number of these will be identified and explained.



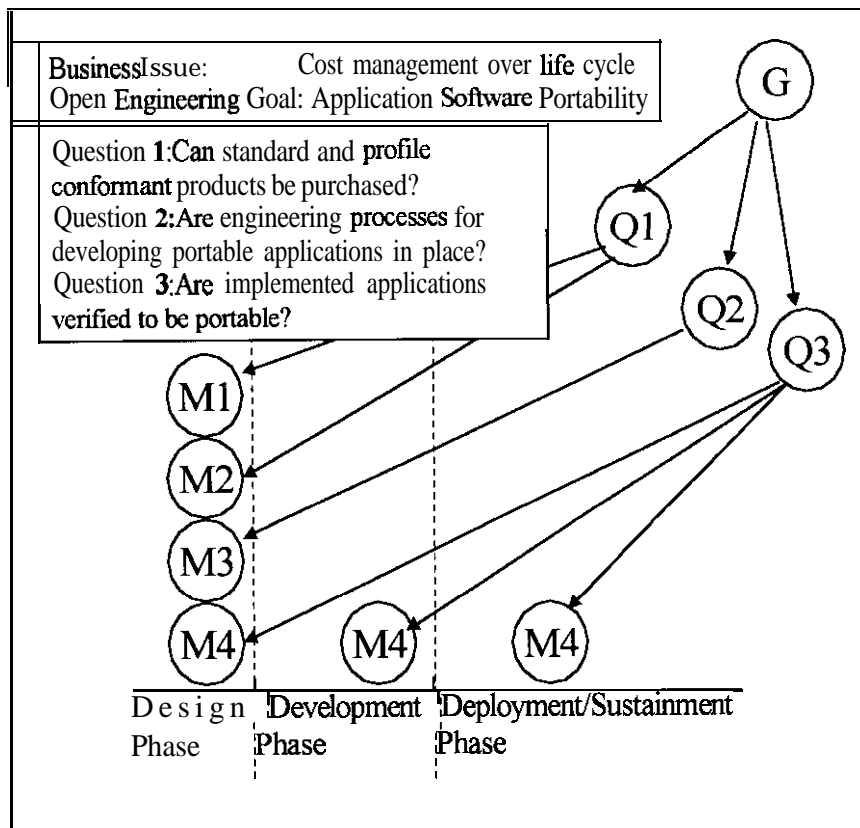


Figure 6 - Phased Measurement Dependency

#### 4.1 Open Systems Engineering Goals and Representative Questions

As has been intimated in the above discussion, an open systems engineering approach:

- Has the potential to contribute to the successful merger of the stability (predictability in performance, reliability, maintainability, . . .) of military systems and the technology innovation of the commercial industry;
- Is an extension of ongoing system engineering efforts, and should not be misconstrued to be an engineering panacea that can be successful devoid of sufficient programmatic and engineering accountability;
- Is based on successful management, process and product related activities;
- Is undertaken in order to achieve economic benefits;
- Is specific to the economic goals and constraints of the system being built and deployed.

This paper has identified business and economic issues that must be addressed. Engineering goals, question and measurement schema is described below. As was discussed above, the measures used within a particular system should be tailored to be

those important to the system being developed. Not all of the goals described below will be appropriate to all systems. The general framework provided, addresses common goals which can be applied to an open system engineering effort. Additionally, when applied to different systems, they may be applied in somewhat different manners, because of the programmatic or development context of the system.

## **4.2 Engineering Goals**

This section describes engineering goals pertinent to the business/economic and engineering issues described in section 3.2. This is not an all-inclusive list at this time, but is used to illustrate the technique for deriving open systems measures used within this report

### **4.2.1 Engineering Goal - Application Portability**

Application portability addresses being able to reduce cost and schedule efforts associated with moving some functionality (usually software) to another host computer platform. Adherence to standards, in the host computer products to which applications will port, and the development of the application software products in ways which strictly adhere to the open standards profiles in use for a system are important to the success of application portability. Application portability is important as underlying host platforms migrate through technology innovation. If the hosts involved both support standards, then the migration should require minimal effort. However if the hosts do not support the standards profiles invoked by the system developer, significant and costly changes may be required. It is expected that over the long life of a military system, the underlying commercial host hardware and software may change, as commercial market trends warrant. A military software application residing on this platform may not need to change, but may not have a choice, but to migrate to the next generation platform. A number of scenarios where underlying platforms change (vendor no longer supports the product; product is found to be unreliable,...) make the cost effectiveness of portability a strong management concern.

Questions/requirements regarding application portability involve:

- The development of open standard profiles based open commercial standards that are appropriate for the system being built;
- The successful identification and purchase of products conforming to the standard and the specific profile of the standard appropriate for the system;
- The development of engineering guidelines, procedures and control processes which support the development of applications which use standard and profile conformant features;
- Testing the application to prove portability.

#### **4.2.2 Engineering Goal - Vendor independence/cost performance choice**

Vendor independence to allow cost performance choices is another key economic goal. It was pointed out earlier that current militarized systems do not provide the opportunity to rapidly insert available commercial technology innovation. In the Navy, the “gray boxes” representing the ANKJYK-7 and AN/UYK-43 line of computers are representative of this inflexibility. In the commercial world, the integration of commercial parts meeting standards allows user to choose the optimal performance-cost choice set for the system being developed. If a component vendor’s products do not meet needed cost performance goals, available alternatives meeting designated standards are chosen. Vendor independence provides a risk mitigation benefit, in that the demise of a particular vendor, or a vendor’s product does not have as significant an impact on a system developer.

Questions/requirements regarding vendor independence involve:

- Ensuring that there are multiple producers of products meeting the standards and your system’s profile of the standards;
- Ensuring that the candidate product vendors supporting/participating in standards development, standards bodies, and independent standards conformance testing bodies?

#### **4.2.3 Engineering Goal - Product, subsystem, system integration; and Component interoperability and compatibility**

A predictable engineering environment is a desirable business goal because it offers credible management insight and therefore forewarning of risks and issues. Where a product or entity is an unknown, it has the potential to impact a schedule in a negative manner. From an engineering perspective, the integration of products, subsystems and systems is often a difficult element. This is especially true of military subsystems. In the commercial product world, products that adhere to standards provide some likelihood of easy integration. However, given that different vendors may properly implement standards in different ways, there may be cases where incompatibilities between standard or profile conformant products exist. Compatibility testing or verification through vendor agreement or through vendor branding can offer some level of assurance that product integration will be predictable. Where a number of subsystems are being developed somewhat independently, and then are to be integrated into a Naval platform for example, compatibility testing can alleviate a significant amount of difficulty subsystem integration. Determining that incompatibilities exist early in the development cycle limits the cost of addressing them. Additionally, the use of standards can mitigate risk associated with a vendor dropping a product line being used within your system. If other

standards based products are available, one layer of integration effort will have been eliminated.

Questions/requirements regarding interoperability and compatibility involve:

- Ensuring that there are multiple producers of products meeting the standards and your system's profile of the standards;
- Ensuring that compatibility and interoperability criteria are used to assess product qualification?

#### **4.2.4 Engineering Goal - Minimize risk associated with commercial product volatility; - Limit transition effort to a new/improved product or to a different vendor**

Commercial products and commercial product vendors are market force driven. Product and product lines may be abandoned. Vendors may go out of business. Users of vendor products may be faced with the transition to another vendor's products. There are open systems engineering activities that can be taken to limit the effort/cost associated with transition to other products.

Questions/requirements regarding transition involve:

- Have alternatives to primary vendor product choices been explored?
- Have actions to limit dependence on a single vendor product or product line been taken?

#### **4.2.5 Engineering Goal - Sustain deployed systems at minimum operational impact; - Provide replacement parts that have no negative impact on operational performance**

Commercial parts that have been integrated into deployed military subsystems represent specific configurations of operationally tested components, verified to function in all of the performance conditions required of the system. The duration of a submarine or surface vessel's need for an operational system is often many years. During that time, various changes may occur to commercial parts used in the system. As the changes occur, they may or may not be announced by a supplier. Consequently, replacement parts with the same part number may operate or be maintained in ways different from the original parts serving that function. Open systems engineering can address parts compatibility for replacement.

Questions/requirements regarding sustaining deployed systems involve:

- Ensuring that there is a parts replacement (component) qualification activity ongoing within a program?

#### **4.2.6 Engineering Goal - Technology insertion for performance improvement; - Limit transition effort to a new technical baseline**

As technology improves, there are opportunities to take advantage of improved performance with a technology area, and to move to a newer or higher performance emergent technology. There are activities and actions appropriate to limit the costs associated with such transition.

Questions/requirements regarding technology insertion involve:

- What activities are ongoing within a program to plan for, identify, and manage the transition to improved implementations of the same technological baseline?
- What activities are ongoing within a program to plan for, identify, and manage the transition to new technological baselines?

### **5.0 Questions and Measures Tables**

The efforts in developing measures to date have resulted in the tables provided within this section. The format of the tables is to identify questions to be asked based on issues, and provide the measurement indicators where they have been identified. This work is expected to be continued, with additional refinement of the effort being published on the POSE home page: “Practical Open Systems Engineering” (POSE) currently maintained by the Naval Undersea Warfare Center Division Newport at URL: <http://arch6.npt.nuwc.navy.mil/pose/>. [2]

The tables provided identify measures in four Open System Architecture areas; Budget Planning, Profiles, Product Selection and Conformance Testing. The first column in each chart provides one or more representative questions for a particular OSA area. Columns 2, 3 and 4 offers an indication of the **category** to which that question provide a measure. Column 5, 6 and 7 shows during which system life cycle phase that question is relevant. Many of the questions have a high importance during the design phase, and progressively less importance as the project moves through the development phase into the sustainment phase. While this characteristic seems to be prominent it is not the only curve, There are other system questions that peak or trough during the development or sustainment phases.

Table nomenclature:

Mgt.	Management
EPc	Engineering Process
Prd	Product
Dgn	Design

Dev.  
sus

Development  
Deployment/Sustainment

Issue	Mgt	EPc	Pro
<b>Budget Planning - Is the Open Systems Engineering effort budgeted for in a reasonable and appropriate manner? Do the associated contracting activities reflect appropriate budget and effort? (ex: 1. Integration contract)</b>			
Have specific open system engineering goals been established for the project? Ex: Which software should be portable? Which subsystems should inter-operate?	X	X	
Has the cost of achieving the goals been established? Has the cost of developing profiles been established? Has the cost of addressing conformance testing been established? Has the cost of finding qualified products been established? Has the cost of developing conformant applications been established?	X	X	
Does the estimated cost of achieving the open system engineering goals meet program constraints and objectives?	X	X	
Is there an understanding of what activities will occur, and approximately when they will occur in the acquisition cycle? Are the activities budgeted for, in the appropriate amounts and at the appropriate program times	X	X	
Technology Identification	X	X	
Standards and standards evolution	X	X	
Profile development	X	X	
Profile application in Product Identification	X	X	
Profile application in Equipment/Implementation conformance	X	X	
Profile application in Application conformance	X	X	
Conformance Qualification Process Established	X	X	
Conformance Qualification Process Employed	X	X	
Technology Refresh	X	X	

Issue	Mgt	EPc	Pr
<b>Profile</b>			
Are the system requirements and interfaces clearly defined, to the extent that they may be mapped against applicable standards, and profiles generated?	X	X	
Does the program/project have continuing access to experts knowledgeable and current on applicable standards?	X	X	
Have profiles been generated and documented?	X		
Is a process in place to maintain (keep current) profiles?	X	X	
Has the impact of using non-standard components been evaluated across the life-cycle?	X		
Is there a policy/process for monitoring/managing non-profile conforming feature use?		X	X
Is there a policy/process for monitoring/managing non-standard feature use?		X	X
Does the developer know where to find publicly tested products addressing his profile?		X	X
Is a current list of proposed/selected components and their level of conformance maintained?		X	X
Does a publicly tested product list, meeting the profile, exist?		X	X
Have all supportability profile requirements been addressed and can they be met?		X	X
Have all environmental profile requirements been addressed and can they be met?		X	X



Issue	Mgt	EPc	Pr
<b>Products</b>			
<p>Is there a process for determining appropriateness of products with respect to a standard/profile?</p> <ul style="list-style-type: none"> <li>• Product qualification</li> <li>• Product generational qualification (same product line – different product generation)</li> <li>• Alternate product qualification (shifting to a different product line)</li> <li>• Are there multiple sources for compliant products?</li> <li>• Are the compliant products interchangeable with others that perform the same functions?</li> <li>• Are the compliant products interoperable?</li> <li>• Do they meet the same performance requirements?</li> <li>• Are the products pin-for-pin interchangeable?</li> </ul>		<p>X</p> <p>X</p> <p>X</p> <p>X</p> <p>X</p> <p>X</p> <p>X</p> <p>X</p>	<p><b>X</b></p> <p><b>X</b></p> <p><b>X</b></p> <p><b>X</b></p> <p><b>X</b></p> <p><b>X</b></p> <p><b>X</b></p> <p><b>X</b></p>
<p>To what extent is an ongoing dialogue being maintained with candidate parts vendors? Early planning can turn risks into opportunities. In an OSE involves</p> <ul style="list-style-type: none"> <li>• comparison of performance requirements to available commercially-based open system interface standards,</li> <li>• determining the relative acceptance in commercial markets,</li> <li>• analyzing and comparing alternative standards and the technologies and product: that implement them for suitability in meeting performance requirements,</li> <li>• anticipating yearly costs for each alternative,</li> <li>• predicting initial and long-term supportability requirements and upgradability, and</li> <li>• ensuring no deviation from open standards.</li> </ul>		<p>X</p> <p>X</p> <p>X</p> <p>X</p> <p>X</p> <p>X</p>	<p><b>X</b></p> <p><b>X</b></p> <p><b>X</b></p> <p><b>X</b></p> <p><b>X</b></p> <p><b>X</b></p>
<p>If an existing system is being modified, is there knowledge and/or understanding of the current state of the standard or other interfaces in use for the system?</p> <ul style="list-style-type: none"> <li>• Have transition paths for subsystems or components been identified?</li> <li>• Have interfaces to legacy systems or components that will not transition been identified?</li> </ul>		<p>X</p> <p>X</p>	<p><b>X</b></p> <p><b>X</b></p>

Will usage of a given open systems product commit us to a single product family? Will such a relationship provide the best value over the life of a system?			
• Is the existing product line support structure well suited to system requirements?		X	X
• Will it be required to supplement or replace existing support products (e.g., technical data, training, repair, upgrade, etc.)?		X	X
• Sharing of common application support resources?		X	X
Additional vendor product/capability considerations;			
• Are vendors capable of OEM repair? What is their history in meeting repair or warranty requirements?		X	X
• Does the current production output meet the program needs?		X	X
• How much of the market share is the system need?		X	X
• Is the program the only reason for a new production run?		X	X
• Are the products real (not vaporware)?		X	X
• Have the product update cycles and support time been defined?		X	X
• If NDI/COTSs are being integrated have the life cycle support times been established?		X	X
• Does the choice of open system interface standards products influence out-year support?		X	X
• Is reliability data available?		X	X
• Are technical manuals available and do they meet program requirements?		X	X
• Are warranties to be provided? Do they meet program needs?		X	X
• Have training considerations been met?		X	X

Issue	Mgt	EPc	Pro
<b>Conformance Testing</b>			
Have the product interfaces been conformance tested?		X	
Does a conformance test capability exist for the product interfaces?		X	
Does the standard(s) have a set of conformance test requirements/procedures?			
Do the conformance test procedures test for all mandatory requirements?		X	
Do the conformance test procedures address the optional and executable requirements?		X	
Does each product have conformance test data available?		X	
Have sources for conformance testing been identified?		X	
Have offered products been tested by independent 3 <sup>rd</sup> party groups?			
Have offered products been verified conformant to required standards?			
Have offered products been verified conformant to required profiles?			
Are the vendors of offered products currently participating in appropriate interface standards bodies?		X	
Have the vendors made or shown any commitment to continuing to follow standards with their product offerings as the standards mature and change?			
Are requirements for derived validation of conformance specified?		X	
What government, contractor and vendor conformance test facilities are in place?		X	
Has conformance testing been conducted on parts/components/LRUs that are in use commercially and which fulfill functional requirement and interface standards?			
Where insufficient or no 3 <sup>rd</sup> party testing is available to validate the conformance of a product to a standard, does the developer have an acceptable approach to demonstrating conformance to required profiles and standards?		X	
Have critical features been identified for which conformance to a standard must be demonstrated?		X	
Have acceptable levels of demonstration been identified, described and agreed upon?		X	

Does the <b>contract/RFP</b> materials require application development processes which are sensitive to the designated open system standards and profiles?	X		
What application guidance-feedback mechanisms for development are required?			
What evidence of applying the application guidance is provided?		X	
Code walkthroughs?		X	
Peer review?		X	
The periodic application of automated code conformance detection tools?		X	
Is guidance provided for application development where non-standard or non-profile conformant features are necessary to achieve functional performance requirements?		X	
Are the specific <b>APIs</b> used by an application documented?		X	
Are demonstrations of portability, interoperability, identified as part of the evidence of application conformance?		X	

## **6.0 Conclusion**

Using COTS based products has become widely accepted within the military systems development community. The use of Open System Engineering and Architecture techniques has not kept pace. There are certain management and engineering disciplines that need to be practiced in order to achieve all the benefits that open systems makes possible.

This paper has introduced three important open systems engineering measurement concepts. The first is the multidimensional nature of an open system engineering effort. Management and engineering processes in addition to product qualification are critical to measure. The second concept is the phased nature of the measurement process. A particular measure has different importance and emphasis within a phase. Progress within a phase is highly dependent on progress in previous phases. Thirdly each program must employ open system engineering measures appropriate to their goals.

## Bibliography:

1. DoD 5000.2R - Subj: Mandatory Procedures for Major Defense Acquisition Programs (MDAP) and Major Automated Information Systems (MAIS) Acquisition Programs - 6 Oct. 1997
2. Practical Open Systems Engineering, Naval Undersea Warfare Center Division Newport, homepage URL: <http://arch6.nt.nuwc.navy.mil/pose/>
3. Secretary of Defense Memorandum, Subject: Specifications & Standards - A New Way of Doing Business, 29 June 1994
4. Under Secretary of Defense for Acquisition and Technology Memorandum, Subject: Acquisition of Weapons Systems Electronics Using Open Systems Specifications and Standards, 29 November 1994
5. Open Systems Joint Task Force home page: <http://www.acq.osd.mil/osjtf/>
6. Joint Technical Architecture home page: <http://www.itsi.disa.mil/>
7. Kowalski, Norman W., Transition *to Open Standards*, Proceedings of the Fifth Systems Reengineering Technology Workshop, February 7-9, 1995, Monterey, CA.
8. Kowalski, Norman W. , *Key Engineering Management Practices to Achieving and Open System in a DoD Environment*, Proceedings of the Eighth Annual Software Technology Conference, April 1996, Salt Lake City Utah
9. Kowalski, Norman W., and Chestnutwood, Mark C., *Transition to Open Standards*, International Test and Evaluation Association (ITEA) 1994, 2nd Annual Test Technology Symposium, Short Course, August 29, 1994, Newport, RI
10. IEEE Std 1003.0-1995, *IEEE Guide to the POSIX Open Systems Environment (OSE)*, IEEE Society New York, NY (December 29, 1995)
11. Quarterman, John S. and Wilhelm, Susanne, UNIX, *POSIX, AND OPEN SYSTEMS*, Addison-Wesley Publishing, Reading, MA (1993)
12. ISO/IEC 9945-1:1996(E) (ANSI/IEEE Std 1003.1 1996 Edition) Information Technology - Portable Operating System Interface (POSIX), Part 1: System Application Program Interface (API) [C Language], 1996
13. Practical Software Measurement, Version 2.1, March 27, 1996, Joint Logistics Commanders Joint Group on Systems Engineering

14. Sage, Andrew, *Systems Management for Information Technology and Software Engineering*, John Wiley and Sons, 1995
15. Chestnutwood, Mark C., and Kowalski, Norman W., *Open Systems Conformance Testing*, Test Technology Transfer Symposium, Dual Use T&E Proceedings, August 29-September 1, 1994, Newport, RI